Projekt dla procesora MIPS - Znajdź znaczniki

Krystian Kamiński - nr indeksu 304013

Implementacja programu znajduje się w pliku znaczniki.asc, a plik testowy jest zmodyfikowanym plikiem ze znacznikami o nazwie input.bmp

Koncepcja algorytmu znajdowania znaczników:

Operując na pikselach z pliku wejściowego, zaczynam sprawdzanie od lewego dolnego rogu, poruszając się w prawo. W momencie osiągnięcia ostatniego piksela w wierszu, przemieszczam się na początek z lewej strony następnego wiersza.
Jeśli piksel nie jest koloru czarnego, poruszam się dalej, w przeciwnym wypadku sprawdzam jego otoczenie z lewej strony i od dołu. Jeśli któryś z tych 2 sąsiadujących pikseli jest koloru czarnego, to oznacza, ze aktualny piksel był już uwzględniony wcześniej. Jeśli nie, to poruszam się w prawo, do momentu przerwania ciągu czarnych pikseli (czy to poprzez wystąpienie innego koloru, albo końca wiersza).
Następnie poruszam się analogicznie w górę, sprawdzając przy okazji kolor pikseli z prawej strony.
Po osiągnięciu końca ciągu czarnych pikseli porównuję długość łańcucha pikseli w pionie i poziomie.
Jeśli zachodzi równość kontynuuję sprawdzanie pikseli znajdujących nie po „wewnętrznej stronie” znacznika.
Podczas sprawdzania kolejnych warstw grubości ramienia, przeglądam czy warstwa jest „spójna”, tzn. czy występują w niej tylko czarny kolor bez wyjątku, albo wszystkie piksele w warstwie są różne od koloru czarnego. Dodatkowo w każdej warstwie następuje sprawdzenie pikseli brzegowych. Punkt przecięcia ramion znacznika zostaje wypisany tylko podczas

wystąpienia w ostatniej pętli wszystkich pikseli o kolorze różnym od czarnego.

Zaimplementowane funkcje:

- start:

Inicjuję tam wartości współrzędnych pikseli oraz licznik punktów

- loop_x:

Pętla iterująca wartość współrzędnej x

- loop_y:

Iteracja wartości współrzędnej y

- start_pixel_in:

Utworzenie kopii wartości pikseli, oraz licznika długości ramion

- check_down_and_move_right:

Sprawdzenie koloru piksel niżej, oraz następnie ruch w prawo

- back_one_position_before_up:

Cofnięcie się o jeden piksel, przed poruszaniem się w górę

- move_up:

Przejście do piksela wyżej, oraz sprawdzenie koloru piksela z prawej strony (jeśli istnieje)

- left:

Inicjalizacja wartości gdy piksel jest umieszczony po skrajnie lewej stronie

- down:

Inicjalizacja wartości gdy piksel jest umieszczony w najniższym wierszu

- down_move_right:

Ruch w prawo, bez sprawdzania koloru piksela poniżej

- compare_length:

Porównanie długości ramion

- check_pixels_inside:

Inicjalizacja zmiennych pomocniczych, sprawdzenie początkowego piksela wewnętrznego

- one_pixel:

Sprawdzenie ostatniego piksela w ostatniej warstwie i jego otoczenia

- check_pixels_inside_loop_right:

Sprawdzenie wszystkich pikseli w danym wierszu w danej warstwie

- check_pixels_inside_loop_up:

Sprawdzenie wszystkich pikseli w danej kolumnie w danej warstwie

- check_over:

Sprawdzenie ostatniego piksela sąsiadującego z pikselem w kolumnie

- last_floor:

Sprawdzenie podstawowych warunków kontynuacji pętli w ostaniej warstwie

- last_floor_continue:

Sprawdzenie wszystkich pikseli w danym wierszu w ostatniej warstwie

- check_pixels_inside_loop_up_last_floor:

Sprawdzenie wszystkich pikseli w danej kolumnie w ostatniej warstwie

- last_pixel:

Sprawdzenie ostatniego piksela sąsiadującego pozostały piksel

- print_sign_point:

Wyświetlenie pożądnych współrzędnych punktu przecięcia ramion

- exit:

Zakończenie działania programu

- read_bmp:

Wczytywanie pliku .bmp

- exit_error:

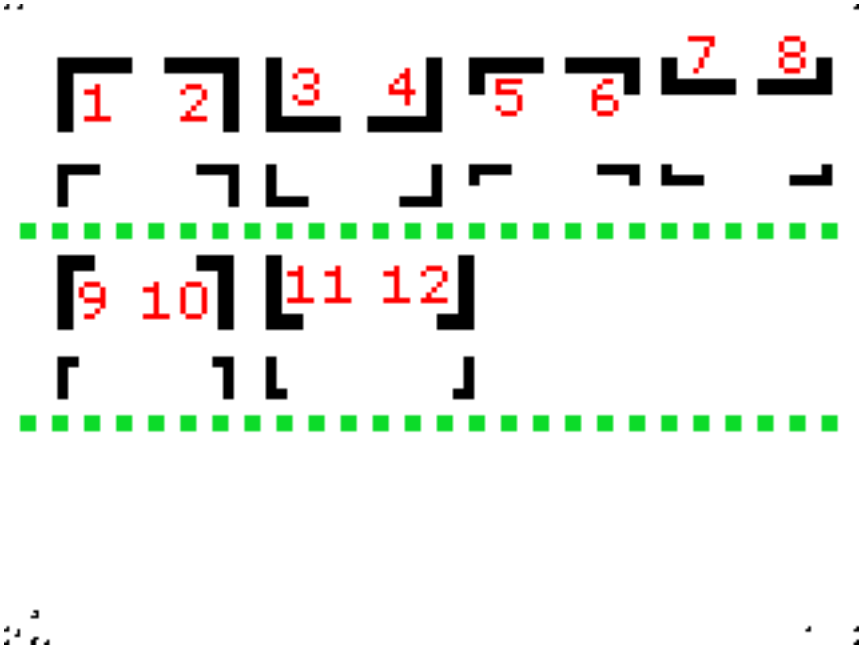
Zakończenie programu zakomunikowanego błędem wczytania pliku

- get_pixel:

Funkcja zwracająca kolor piksela

Testowanie:

W pliku input.bmp, który jest zmodyfikowanym plikiem example_markers.bmp umieściłem dodatkowe znaczniki oraz znaczniki niepoprawne, w różnych miejscach niezbędnych do wykazania pełnej skuteczności działania programu.



Na powyższym zdjęciu znajduje się łącznie 11 poprawnych znaczników nr 4.
Są nimi:

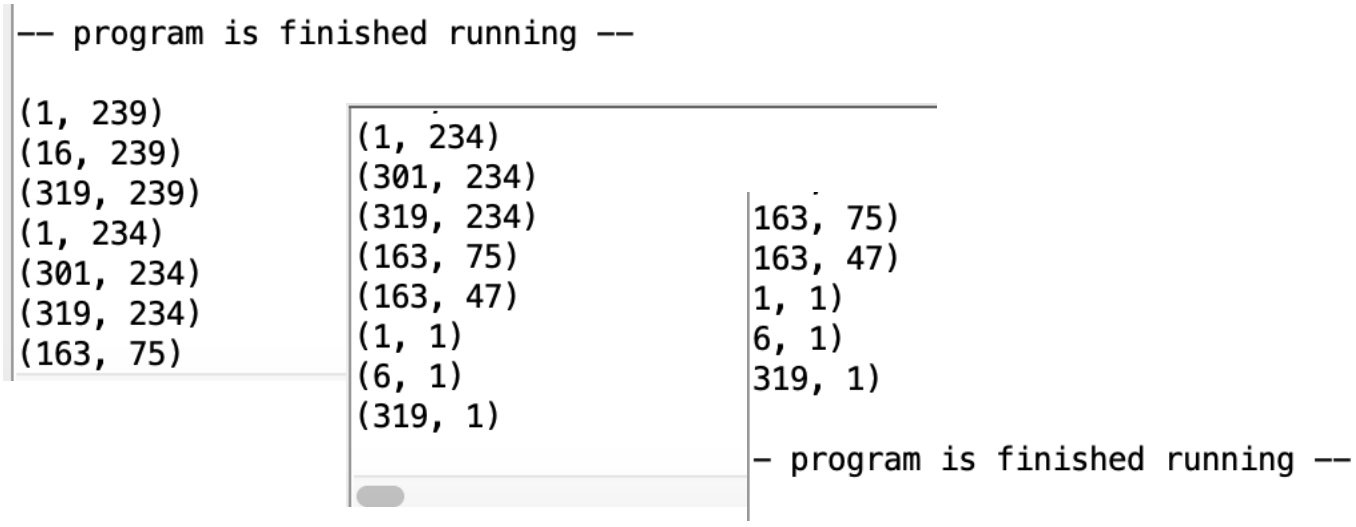
- Znacznik sąsiadujący z lewą i górną ścianą,
- Znacznik sąsiadujący tylko z górną ścianą,
- Znacznik sąsiadujący z prawą i górną ścianą,
- Znacznik sąsiadujący tylko z prawą ścianą,
- Znacznik sąsiadujący z dolną i prawą ścianą,
- Znacznik sąsiadujący tylko z dolną ścianą,
- Znacznik sąsiadujący z lewą i dolną ścianą,
- Znacznik sąsiadujący tylko z lewą ścianą,

- Znacznik nie sąsiadujący z żadną ścianą (w okolicach lewego dolnego rogu)
- Oraz dwa znaczniki które były już obecne w pliku example_markers.bmp, o grubości większej niż 1

Z całego pliku program wypisuje prawidłowo tylko 11 punktów przecięcia ramion należących właśnie do wyżej przedstawionych znaczników.

Dodatkowo utworzone zostały znaczniki pułapki, które posiadają, np. dodatkowy, nie dozwolony piksel, przez który figura nie może być znacznikiem.

Testy można łatwo powtórzyć, poprzez uruchomienie programu z pliku znaczniki.asc



Rezultat działania programu, wypisującego pożądane punkty (Wynik znajduje się na 3 zdjęciach, przez brak możliwości rozciągnięcia Outputu)